

UNITED STATES PATENT APPLICATION

FOR

METHOD OF SELECTIVELY COMPRESSING DATA PACKETS

INVENTORS:

NANJUNDIAH VISWANATH

PREPARED BY:

CHARLES K. YOUNG
INTEL CORPORATION
2200 MISSION COLLEGE BLVD.
SANTA CLARA, CA 95052-8119

(408) 765-8080

Attorney's Docket No. 42390.P10198

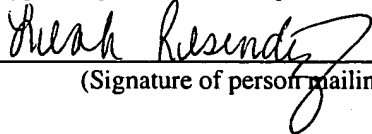
"Express Mail" mailing label number EL627464278US

Date of Deposit December 15, 2000

I hereby certify that this paper or fee is being deposited with the United States Postal Service
"Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and
is addressed to the Assistant Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Leah Resendez

(Typed or printed name of person mailing paper or fee)



(Signature of person mailing paper or fee)

METHOD OF SELECTIVELY COMPRESSING DATA PACKETS

BACKGROUND OF THE INVENTION

5 1. Field of the Invention

The described invention relates to the compression of network data packets.

2. Description of Related Art

10 In network communications, data is typically compressed to speed up communications over connections having a low bandwidth. However, compression of data takes up resources and time. If one tries to compress data that is already compressed, the compression typically fails, and a larger data item than the original often results.

15 Figure 1 shows a prior art example of a virtual private network that allows two gateways 10 and 20 to talk to each other over the Internet 30. The communications are encrypted so that any unauthorized interception of data over the public Internet will not be decipherable.

20 Multiple computers 11-14 are connected to the first gateway 10 and multiple computers 21-23 are connected to the second gateway 20. The gateways 10 and 20 communicate with each other by sending data packets. The data packets are compressed and encrypted on one gateway then sent over the Internet 30 to the other gateway which decrypts the data packets and decompresses them.

IPsec (latest revision, RFC 2401, November 1998), an industry specification published by the Internet Engineering Task Force, provides security at the Internet

Protocol level. Compression is performed before encryption of data packets.

Compression avoids fragmentation of data and increases throughput. When compression is successful (i.e., it reduces a data packet's size), a new header is added to the data packet. This header is used by the decoding side to apply the correct
5 decompression algorithm after the decryption process. More information on IPsec can be found on its web site www.ietf.com.

Usually two peers negotiate via an Internet Key Exchange ("IKE") process whether to use compression or not. When the type of traffic is primarily text data, compression is turned on. When the traffic is primarily compressed, e.g.,
10 compressed video only, compression is turned off since there is no need to do compression (it is already compressed). However, for a mixture of compressed data and uncompressed data (e.g., text data) such as in a web page, the decision whether to compress is not clear.

One way of addressing the problem is a heuristic approach in which data
15 packets are sampled periodically. If the current data packet is compressed, then compression is turned off for a certain period of time. Subsequent data packets are sampled, and when the data packets are no longer compressed, then compression is turned off. However, such tests for compression take time and resources.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a prior art example of a virtual private network that allows two gateways to talk to each other over the Internet.

Figure 2 is a flowchart showing one embodiment of selectively compressing
5 data packets.

Figure 3 is a flowchart showing an embodiment for determining whether to bypass compression.

10

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

DETAILED DESCRIPTION

Selectively compressing data to avoid trying to compress data packets that are already compressed saves valuable cycles. Searching for a predetermined marker in the data packet is often less expensive in terms of time and resources than
5 the compression process.

Figure 2 is a flowchart showing one embodiment of selectively compressing a data packet. First, a determination is made whether compression is enabled (box 100). As previously described in the background section, whether compression is enabled is often decided in an Internet Key Exchange ("IKE") process. If
10 compression is enabled then execution continues at box 102. If compression is not enabled, then execution continues at box 190.

At box 102, a determination is made whether to bypass the compression process, as will be described in detail with respect to Figure 3. If the determination is that the compression process is bypassed then execution continues to block 190.
15 If there is no bypass, execution proceeds from block 102 to block 104, at which the data packet is compressed. After compression, execution proceeds from block 104 to block 190.

At block 190, further processing of the packet data is performed. In one embodiment, the data packet is encrypted and then forwarded over a network.

20 Figure 3 is a flowchart showing an embodiment for determining whether to bypass compression. As each data packet is processed, the flowchart progresses similar to a state diagram, and a variable BYPASS indicates whether to bypass the compression or not. The flowchart starts at block 200, at which BYPASS is

initialized to FALSE. Any data packet forwarded while BYPASS is FALSE will go through the compression process. The flowchart continues at block 202 at which the data packets are searched for a start marker that indicates that compressed data follows.

5 As one example, a GIF image, which is a compressed image, includes in a header the text string "GIF". A search for this string can be performed on each of the data packets. In one embodiment, a search string engine of a network processor can be used to search the data packets. These search string engines can quickly detect the presence of such a string. Other types of compressed data have other
10 markers which indicate the beginning of the compressed data, whether it be compressed audio, video, graphic or other types of data.

 At block 204, if the start marker is not found then the search for the start marker continues (back to block 202). Once the start marker is found, execution proceeds at block 206, at which BYPASS is set to TRUE. Any data packets
15 processed while BYPASS is TRUE will bypass the compression process.

 At block 208, data packets are searched for an end marker that indicates the end of compressed data. As an example, the end of a GIF image is signified by an end marker of a string having two consecutive bytes of "0" followed by ";" at the end of a data packet. If the end marker is not detected at block 210, then control
20 proceeds to block 220 at which a timeout decision is made. The timeout allows for the bypass to be suspended if the end marker has not been found within a certain limit, such as within a particular number of data packets, bytes, or time limit. At

block 220, if there is no timeout, then the search for the end marker continues at block 208.

However, if there is a timeout, e.g., due to the end marker not being found within 4K bytes, then the timeout is reset and control proceeds to block 222. At
5 block 222, a test is performed to determine whether the current data packet being processed is compressed. One way to detect compression is to try to compress the data block. If the compression fails, i.e., results in a larger packet size, then the data packet is already compressed. If the test shows that the current data packet being processed is compressed, then control proceeds to block 208 to continue searching
10 for the end marker. However, if the test shows that the current data packet being processed is not compressed, then somehow the end marker was missed. Process control proceeds from block 222 back to block 200. Similarly, at block 210, when the end marker is found, then process control continues at block 200, and the process starts over.

15 In one embodiment, the search for start markers (block 202) and end markers (block 208) is performed on only portions of each data packet. As one example, certain headers, such as the IP-header and TCP-header, are excluded from the search.

In one embodiment, both the start marker and the end marker may be in the
20 same data packet. In this case, that data packet bypasses the compression process, and compression resumes until another start marker is found.

Of course, the described embodiment can easily be expanded to accommodate more than one type of start and end marker. For example, there are

many different types of compressed data, and a search can be made for multiple start markers. Once one of the start markers is found, then the appropriate type of end marker can be searched for. Overlapping types of start and end markers can also be accommodated.

5 In one embodiment, software for programming a computer to operate as described can be provided as instructions stored on floppy disk, CD-ROM, or other storage media. Alternatively, the software can be downloaded via the Internet or a wireless network. The software is then installed to a storage medium on the host system, such as a hard disk, random access memory, or non-volatile memory.

10 Thus, a method for selectively compressing data packets is disclosed. The specific arrangements and methods described herein are merely illustrative. Numerous modifications in form and detail may be made without departing from the scope of the invention as claimed below. The invention is limited only by the scope of the appended claims.